IN THE CLAIMS

Please amend the claims as follows.

1. (Previously Presented) A method of sharing secure cryptographic connections between trusted computing entities which share a secret value, the computer-implemented method comprising the steps of:

connecting an originally-connected entity to an original endpoint, the originally-connected entity having an entity name and cryptographic context information; and

creating an entity identifier by encoding the entity name and the secret value such that by using the secret value information necessary to access the cryptographic context information can be retrieved, and wherein the entity identifier acts as an index into a data structure for acquiring the cryptographic context information.

2. (Original) The method of claim 1, further comprising the step of passing the entity identifier to at least one subsequently-connected computing entity which seeks to connect to the original endpoint.

3. (Original) The method of claim 2, further comprising the step of decoding the entity identifier using the secret value, thereby determining information necessary to access the cryptographic context information.

4. (Original) The method of claim 3, wherein the step of decoding the entity identifier comprises using the secret value as a key to an encryption algorithm to decrypt the entity identifier.

5. (Original) The method of claim 3, wherein there is at least one other trusted computing entity, the trusted computing entity possessing a trusted entity name, and the decoding step comprises encoding at least one trusted computing entity name and the secret value to produce a computed identifier, and then comparing the computed identifier to the entity identifier to determine if they match.

6. (Original) The method of claim 3, further comprising a deconcatenating step which deconcatenates a random number from the entity identifier prior to the decoding step, and the decoding step uses the random number, a trusted entity name from one of the group of trusted entity names and the secret value to produce a computed identifier and then compares the computed identifier to the entity identifier to determine if they match.

7. (Original) The method of claim 6, wherein the computed identifier and the entity identifier do not match and wherein there is at least one other trusted computing entity, further comprising repeating the decoding step until a match is found or until there are no more trusted computing entities to try.

8. (Original) The method of claim 2, wherein the subsequently-connecting computing entity uses the originally-connected entity name to access the originally-connected entity cryptographic context information, and the subsequently-connecting computing entity uses the originally-connected entity cryptographic context information in a secure connection to the original endpoint.

9. (Original) The method of claim 1, whereby the creating step comprises using a hash function with an input and an output, said input comprising the entity name and the secret value, said output comprising the entity identifier.

10. (Original) The method of claim 1, whereby the creating step comprises using a hash function with an input and an output, said input comprising a bitwise concatenation of the entity name, the secret value, and a random number, said output of the hash function being at least bitwise concatenated with the random number.

11. (Original) The method of claim 10, wherein the hash function is uninvertible.

12. (Original) The method of claim 10, wherein the hash function is SHA-1.

13. (Original) The method of claim 1, wherein the creating step comprises using an encrypting algorithm that uses a key to encrypt the entity name using the secret value as the key, the encrypted entity name comprising the entity identifier.

14. (Original) The method of claim 1, wherein the creating step comprises bitwise concatenating the entity name and a random identifier comprising a result and then using an encrypting algorithm that comprises an input, a key, and an output, whereby the result comprises the input, the secret value comprises the key, and the output comprises the entity identifier.

15. (Original) The method of claim 14, wherein the encrypting algorithm is Triple DES.

16. (Original) The method of claim 2, wherein the originally-connected entity is no longer connected to the original endpoint.

17. (Previously Presented) A system for sharing secure cryptographic connections, the system comprising: an originally-connected trusted entity which comprises an originally-connected entity name and cryptographic context information; at least one other trusted entity, which comprises another entity name; a secret value known to the at least two trusted entities; and a connection identifier comprising an encoded version of the originally-connected entity name and the secret value, and wherein the connection identifier is linked to the cryptographic context information.

18. (Original) The system of claim 17, further comprising an connection identifier passer which passes the connection identifier to the at least one other trusted computing entity which seeks to connect to the original endpoint.

19. (Original) The system of claim 18, further comprising a connector which uses the connection identifier to access the originally-connected entity cryptographic context information, and which uses the originally-connected entity cryptographic context information to establish a secure connection to the original endpoint.

20. (Original) The system of claim 19, wherein the originally-connected entity is no longer connected.

21. (Original) The system of claim 17, further comprising a decoder which returns the originally-connected entity name when it is given the connection identifier.

22. (Original) The system of claim 21, wherein the decoder decrypts the connection identifier into an intermediate value when given the secret value and then deconcatenates the originally-connected entity name and the random id from the intermediate value.

23. (Original) The system of claim 21, wherein the decoder deconcatenates the connection identifier into an intermediate value and a random number, and wherein the system further comprises a recoder which recodes the random number, the at least one other trusted entity name, and the secret value into a test identifier.

24. (Original) The system of claim 23, further comprising a tester which compares the connection identifier with the test identifier, and if they are equal determines that the trusted entity name used by the recoder is the originally-connected trusted entity name, and if they are not equal chooses a previously-unchosen trusted entity name as input into the recoder.

25. (Original) The system of claim 17, further comprising an encoder which encodes the connection identifier using at least the originally-connected entity name and the secret value.

26. (Original) The system of claim 25, whereby the encoder bitwise concatenates the entity name and a random number producing an intermediate value and then uses an encryption algorithm that takes a key to encrypt the intermediate value using the secret value as the key.

27. (Original) The system of claim 25, whereby the encoder comprises an encryption algorithm.

28. (Original) The system of claim 27, wherein the encryption algorithm comprises symmetric key encryption.

29. (Original) The system of claim 27, wherein the encryption algorithm comprises public key encryption.

30. (Original) The system of claim 27 wherein the encryption algorithm comprises Diffie-Hellman key exchange encryption.

31. (Original) The system of claim 25, whereby the encoder comprises a hash function.

32. (Original) The system of claim 25, whereby the encoder creates the connection identifier by bitwise concatenating two values; the first value being a random number, and the second value being the output of a hash function with an input and an output, the input comprising the bitwise concatenation of the entity name, the secret value, and the random number.

33. (Original) The system of claim 25, whereby the encoder creates the connection identifier by using a key-dependent hash with an input and a key, with the input comprising the originally-connected entity name, and the key comprising the secret value.

34. (Previously Presented) A signal embodied in a computer, the signal comprising an entity identifier which comprises an encoded version of an entity name, a secret value, and a random number, and wherein the encoded version of the entity name, the secret value and the random number are adapted to be bitwise concatenated with one another to produce an intermediate value, the intermediate value is adapted to be hashed to acquire a hash result, the hash result is adapted to be bitwise concatenated with the random number to produce the entity identifier.

35. (Previously Presented) A configured storage medium embodying data and instructions readable by a computer to perform a method of sharing secure cryptographic connections between trusted computing entities which share a secret value, the computer-implemented method comprising the steps of:

connecting an originally-connected entity to an original endpoint, the originally-connected entity having an entity name and cryptographic context information; and

creating an entity identifier by encoding the entity name and the secret value, such that by using the secret value information necessary to access the cryptographic context information can be retrieved, and wherein the entity identifier serves as an index into a data structure for acquiring the cryptographic context information.

36. (Original) The configured storage medium of claim 35, whereby the creating step comprises using a hash function.

37. (Original) The configured storage medium of claim 35, wherein the creating step comprises encrypting a bitwise concatenation of the entity name and a random value.